

1 MTfit: A Bayesian approach to seismic moment 2 tensor inversion

3

4 D.J. Pugh^{1,2,3*}, R.S. White¹

5

6

7 ¹Bullard Laboratories, Department of Earth Sciences, University of Cambridge, CB3 0EZ, UK

8 ²Schlumberger Gould Research Centre, High Cross, Cambridge, UK

9 ³Now at McLaren Applied Technologies, McLaren Technology Centre, Chertsey Road, Woking,
10 UK

11

12 *Email:david.j.pugh@cantab.net

13

14 *For publication in SRL: Electronic Seismologist.*

15

16 In original form 13th December 2017

17 Abstract

18 MTfit is a Python module for Bayesian moment tensor source inversion of earthquake seismic
19 data using polarities, amplitudes or amplitude ratios. It can solve for double couple or full
20 moment tensor solutions, taking into account uncertainties in polarities, take-off angles of the
21 rays from the source to the receiver, and amplitudes. It provides an easily accessible and
22 extendable approach to earthquake source inversion which is particularly useful for local and
23 regional events.

24 Introduction

25 Earthquake source inversion is carried out at many seismological observatories and research
26 facilities around the world. Pugh *et al.* (2016b) introduced a Bayesian approach to estimating
27 the moment tensor of the source using polarities and amplitude ratios, which was extended to
28 include automated Bayesian polarity probability estimates by Pugh *et al.* (2016a). This
29 approach differs from existing approaches, such as FPFIT (Reasenbergs & Oppenheimer, 1985),
30 HASH (Hardebeck & Shearer, 2002, 2003) and FOCMEC (Snoke, 2003), because it uses
31 polarities and amplitude ratios in a Bayesian framework to estimate the full source probability
32 density function (PDF) for the double-couple and full moment tensor model spaces. The
33 approach can include location and velocity model uncertainties, as well as marginalizing over
34 measurement uncertainties in the data.

35 The approach of Pugh *et al.* (2016b) has been developed into MTfit, a Python package for
36 source inversion. Python is a common programming and scripting language with many scientific
37 modules available, both for mathematical calculation such as NumPy (<https://www.numpy.org>)
38 and SciPy (<https://www.scipy.org/>), and for seismological applications such as ObsPy

39 (<https://www.obspy.org>) (Beyreuther *et al.*, 2010).

40 Python and many of its modules are open source, allowing easy code development and removing
41 licensing restrictions. Moreover, Python is platform independent, intuitive, and accessible, with a
42 good shell interface in the form of iPython (<https://ipython.org/>). It is used in many fields and
43 is easy to install on almost any computer platform. Python can also interface easily with C and
44 Fortran libraries, and can call functions from compiled C modules, such as those generated with
45 Cython (<http://cython.org/>), with no difference from normal Python functions. Note that earlier
46 versions of the code were referred to as MTINV, but the name has been changed to MTfit to
47 avoid a clash with a previous use of the name MTINV. MTfit has already been used in several
48 studies, including these reported by Wilks *et al.* (2015), Greenfield & White (2015), Pugh *et al.*
49 (2016b), Schuler *et al.* (2016), Mildon *et al.* (2016), Smith *et al.* (2017) and Hudson *et al.* (2017).

50 In this paper, the functionality of MTfit is introduced, and examples of the approach are shown.
51 The model probability estimates derived from the Bayesian evidence are explored, and methods
52 of extending MTfit are presented. Lastly, two examples of plotting the results from MTfit are
53 shown. A flow diagram outlining the main modules of MTfit is shown in Figure 1.

54

55 **Moment Tensor Inversion**

56 MTfit uses the Bayesian source inversion approach from Pugh *et al.* (2016b). The solutions are
57 estimated using polarities and amplitude ratio data, although the code is extendible, so it is
58 possible to include other data types in this framework. MTfit incorporates uncertainty estimates
59 both in the data, such as those arising due to noise, and due to the model (and location), in the
60 resultant posterior PDF. We have developed three sampling approaches, each with different
61 advantages and disadvantages (Pugh 2015). MTfit can also be used for relative amplitude

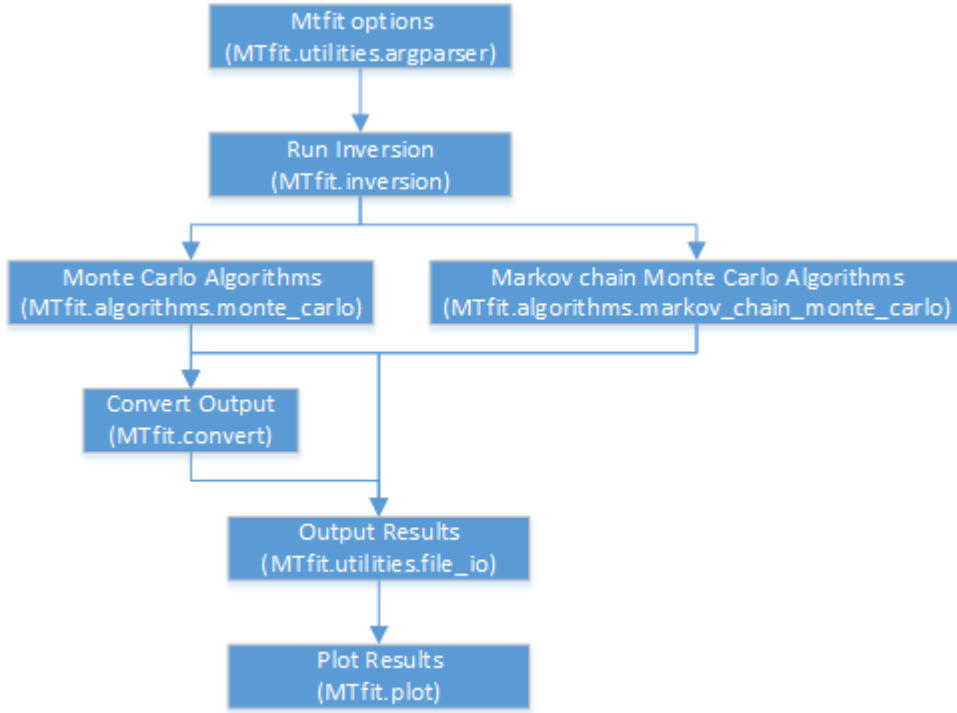


Fig. 1: Flow diagram outlining the main steps in the moment tensor inversion package.

62 inversions (Pugh 2015).

63 The MTfit approach evaluates the data likelihood ($p(data|model)$) for the observations and
 64 measurement uncertainties at each receiver over a range of random moment tensor samples. These
 65 likelihoods are combined to produce the likelihood for all the receivers. Location and model
 66 uncertainties are included by generating samples of locations of the receivers on the focal sphere,
 67 corresponding to the distribution of possible locations of the earthquake, which are marginalized
 68 over to produce the location marginalized likelihood. The resultant likelihood is then saved. If a
 69 Markov chain algorithm is used, the moment tensor samples are generated and saved using the
 70 Markov chain algorithm.

71 MTfit can be called both from the command line and from within the Python interpreter. On
 72 the command line:

73 `$ MTfit event_data.inv`

74 is equivalent to

75 >>> import MTfit

76 >>> MTfit.MTfit(data_file="event_data.inv")

77 in the Python interpreter.

78 Three search algorithms have been implemented. The simplest is a Monte Carlo (MC) random
79 sampling algorithm, which can be limited either by the number of samples or by the elapsed time
80 (in seconds):

81 \$ MTfit --algorithm=iterate --max-samples=100000 event_data.inv

82 \$ MTfit --algorithm=time --max-time=600 event_data.inv

83 The other two algorithms are Markov chain Monte Carlo (McMC) approaches: Metropolis-
84 Hastings McMC and reversible jump McMC. These are described in detail in Pugh (2015). The
85 two McMC algorithms can be selected on the command line:

86 \$ MTfit --algorithm=mcmc --chain-length=100000 event_data.inv

87 \$ MTfit --algorithm=transdmcmc --chain-length=100000 event_data.inv

88 MTfit can be constrained to the double-couple space or allowed to explore the full moment
89 tensor space. This also allows comparisons to be made between the different models and can
90 be used to evaluate the model probabilities. Additional sampling algorithms can be added using
91 entry points. The prior distribution for generating the source models can also be changed, either
92 to select specific submodels or to change the prior distribution on the source model. An example
93 of the former is the strike-slip example in MTfit.extensions.model_sampling_strike_slip, which
94 generates only strike-slip sources rather than full double-couple sources.

95 The full moment tensor space used in the calculation has 5 free parameters (the 6 parameters
96 from the symmetrical moment tensor normalised to 1 because the data types cannot constrain the
97 seismic moment).

98 There are several different output formats, including a MATLAB® format and a format
99 based on the .hyp format of NonLinLoc (Lomax *et al.*, 2000, 2009), with a binary structure for
100 the moment tensor samples, and it is easy to extend the output formats using the entry points
101 described below.

102 A Simple Example

103 This example shown in Figure 2, using real data collected from the Krafla volcano in northern
104 Iceland can be found at
105 https://github.com/djpugh/MTfit/tree/master/examples/SRL_examples/krafla.py. It is a
106 strongly non-double-couple event, with manually picked P- and S-wave arrival times and P-wave
107 polarities, located using NonLinLoc (Lomax *et al.* 2000, 2009). In this case, it is difficult to
108 measure the amplitudes of the S-wave arrivals, so amplitude ratios are ignored. Instead,
109 polarities and polarity probabilities (Pugh *et al.*, 2016a) are used separately to constrain the
110 source, along with the location data. This event is shown in Pugh *et al.* (2016b) and investigated
111 in more detail in Mildon *et al.* (2016), and has large location uncertainty, especially in the
112 take-off angle of the source-to-receiver arrays (Figure 2). The script used for generating Figure 2
113 is equivalent to outputting the data file and location uncertainty from Python:

```
114 >>> from MTfit.examples.example_data import krafla_event, krafla_location
115 >>> data = krafla_event()
116 >>> open('krafla_event.scangle', 'w').write(krafla_location())
117 >>> import pickle
118 >>> pickle.dump(data, open('krafla_event.inv', 'wb'))
```

119 and calling MTfit with the command line options:

```
120 $ MTfit --location_pdf_file_path=krafla_event.scangle --algorithm=iterate
```

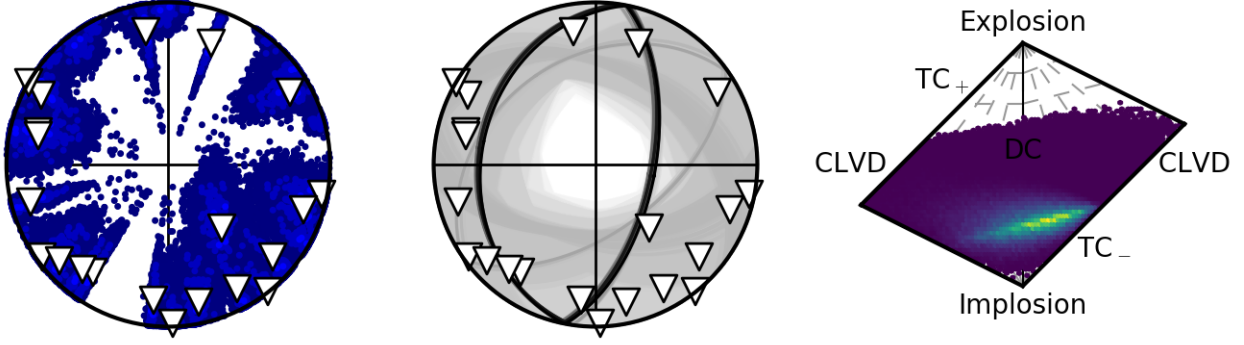


Fig. 2: Krafla example results from the script at https://github.com/djpugh/MTfit/tree/master/examples/SRL_examples/krafla.py (run with 1,000,000 samples). The first plot shows the station distribution of observed receivers on the focal sphere, all with negative polarity, determined from the NonLinLoc estimate of the location PDF. The lighter points correspond to more likely receiver locations, and the maximum likelihood station locations with observed polarities are shown as triangles. The second plot shows the fault plane distribution for the double-couple constrained solution, with darker fault planes more likely. The last plot shows the Hudson type plot of the marginalized source-type PDF from the full moment tensor solution, with dark regions corresponding to low-probability source-types and lighter areas to higher probability types.

```

121  --pmem=1 --double-couple --max-samples=
122  --inversion-options=PPolarity --convert --bin-scatangle krafla_event.inv
123  $ MTfit --location_pdf_file_path=krafla_event.scatangle --algorithm=iterate
124  --pmem=1 --max-samples=10000000 --inversion-options=PPolarity --convert
125  --bin-scatangle krafla_event.inv

```

126 It is possible to run these inversions using other algorithms, such as those described in Pugh
127 (2015), as described in the MTfit documentation.

128 The inversion also produces distributions of the moment tensor parameters which can be
129 plotted using the MTplot command to show the distribution of individual parameters (Figure 3).

130 Model Probabilities

131 Pugh *et al.* (2016b) introduced a method of estimating the model probabilities using the
132 Bayesian evidence. MTfit can include the Bayesian evidence estimation required for this

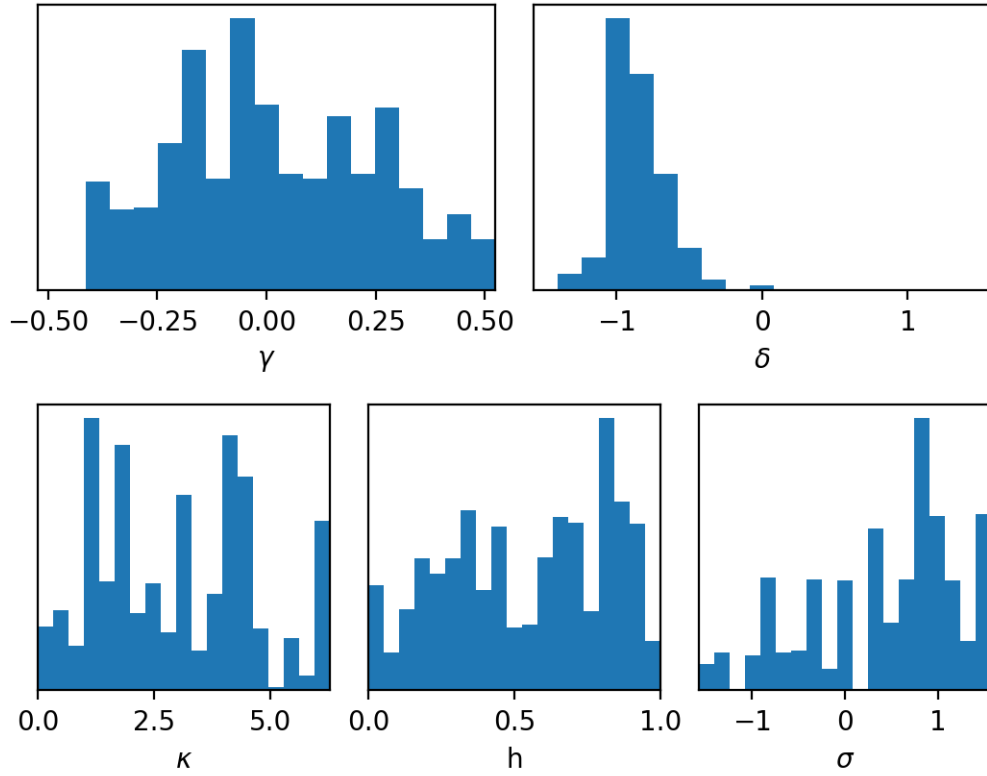


Fig. 3: Marginalised posterior parameter distribution histogram for the five parameters described in Tape & Tape (2012) for the event shown in Fig. 2. γ and δ describe the moment tensor pattern, while κ (strike angle), h (cosine of dip) and σ (rake) describe the orientation. All parameters are dimensionless except κ and σ , which are in radians. This shows that the distributions are well constrained for the δ component, but are less well constrained for the fault plane orientation and γ component.

133 calculation in its results. To estimate the model probabilities for the double-couple and full
 134 moment tensor models, it is necessary to run the inversions in both the model spaces. The
 135 `-double-couple` command line flag will constrain the model to the double-couple space; otherwise
 136 the full moment tensor space is used. The Bayesian evidence values generated by each inversion
 137 can be combined and normalized to produce the model probabilities

$$\ln(\mathcal{B}_{\max}) = \max(\ln(\mathcal{B}_{\text{DC}}), \ln(\mathcal{B}_{\text{MT}})), \quad (1)$$

$$p_{\text{DC}} = \frac{e^{\ln(\mathcal{B}_{\text{DC}}) - \ln(\mathcal{B}_{\max})}}{e^{\ln(\mathcal{B}_{\text{DC}}) - \ln(\mathcal{B}_{\max})} + e^{\ln(\mathcal{B}_{\text{MT}}) - \ln(\mathcal{B}_{\max})}}, \quad (2)$$

$$p_{\text{MT}} = \frac{e^{\ln(\mathcal{B}_{\text{MT}}) - \ln(\mathcal{B}_{\max})}}{e^{\ln(\mathcal{B}_{\text{DC}}) - \ln(\mathcal{B}_{\max})} + e^{\ln(\mathcal{B}_{\text{MT}}) - \ln(\mathcal{B}_{\max})}}, \quad (3)$$

138 where \mathcal{B} corresponds to a Bayesian evidence estimate (MTfit outputs the logarithm of the
 139 Bayesian evidence estimate) and p_{DC} and p_{MT} correspond to the double-couple and full moment
 140 tensor model probabilities respectively. As MTfit can be extended (see below), it is possible to
 141 introduce new model constraints, and the model probabilities can be extended using a similar
 142 logic to that in Eqs 1 – 3. For the example shown in Figure 2, the p_{DC} estimate is 0.0008, and the
 143 p_{MT} estimate is 0.9992. This can be calculated using the `MTfit.probability.model_probabilities()`
 144 function, which takes the calculated logarithm of the Bayesian evidence estimates as arguments.

145 Alternatively, the model probability can be estimated using the transdimensional (reversible
 146 jump) MCMC algorithm, selected using `-algorithm = transdmcmc`. This algorithm uses the
 147 reversible-jump approach described in Pugh (2015). The model probability estimates from this
 148 algorithm are consistent with those from the Bayesian evidence estimators (Pugh, 2015), and
 149 both estimates can be used as a hypothesis test for whether or not the source is double-couple.

150 Figure 4 shows inversions for a synthetic double-couple source with a range of different signal to
 151 noise ratios (SNR) and polarity picks. As the SNR decreases, fewer picks can be made on

arrivals, thus reducing the constraints available for fitting. The two left hand columns show the
 results using only polarity picks, while the two right hand columns include constraints from
 polarity and amplitude data. We show the solutions if they are constrained to be double-couple
 in the first and third columns. The constraints also allowed full moment tensor solutions to be
 calculated, and these are shown in the second and fourth columns. It is clear that, as expected,
 the solutions are constrained better for the higher SNR cases. But there is a marked
 improvement in the constraints if amplitude ratios as well as polarity data are also taken into
 account (third and fourth columns in Figure 4). Indeed, for the better SNR cases, down to SNR
 of 3, the moment tensor solutions that include amplitude ratios still return a double-couple
 solution as the best fit, and even with a SNR of 2, the best solution is close to a double couple:
 these full moment tensor solutions also faithfully reproduce the strikes and dips of the nodal
 planes of the synthetic example we used (top row, Figure 4), at least down to SNR as low as 3.

Computer Run Times

Typical run times depend on the sampling size and the chosen algorithm as well as details of the
 particular moment tensor solution. Figure 5 shows processor elapsed time for calculation of a
 typical double couple source mechanism using a relatively slow single core computer. The
 random sampling and McMC algorithms produce comparable results, but the McMC calculation
 takes about 5 times longer to achieve similar resolution. Random sampling requires typically 50
 million samples to produce a good sampling of the PDF, though the peak is sharpened if the
 number of samples is increased to 500 million. The McMC approach requires far fewer samples
 than random sampling, with a chain length of 50,000 for the McMC approach giving comparable
 results to 100 million random samples. However, the calculation of the likelihood for a large
 number of samples is much faster with the random sampling algorithm because the McMC

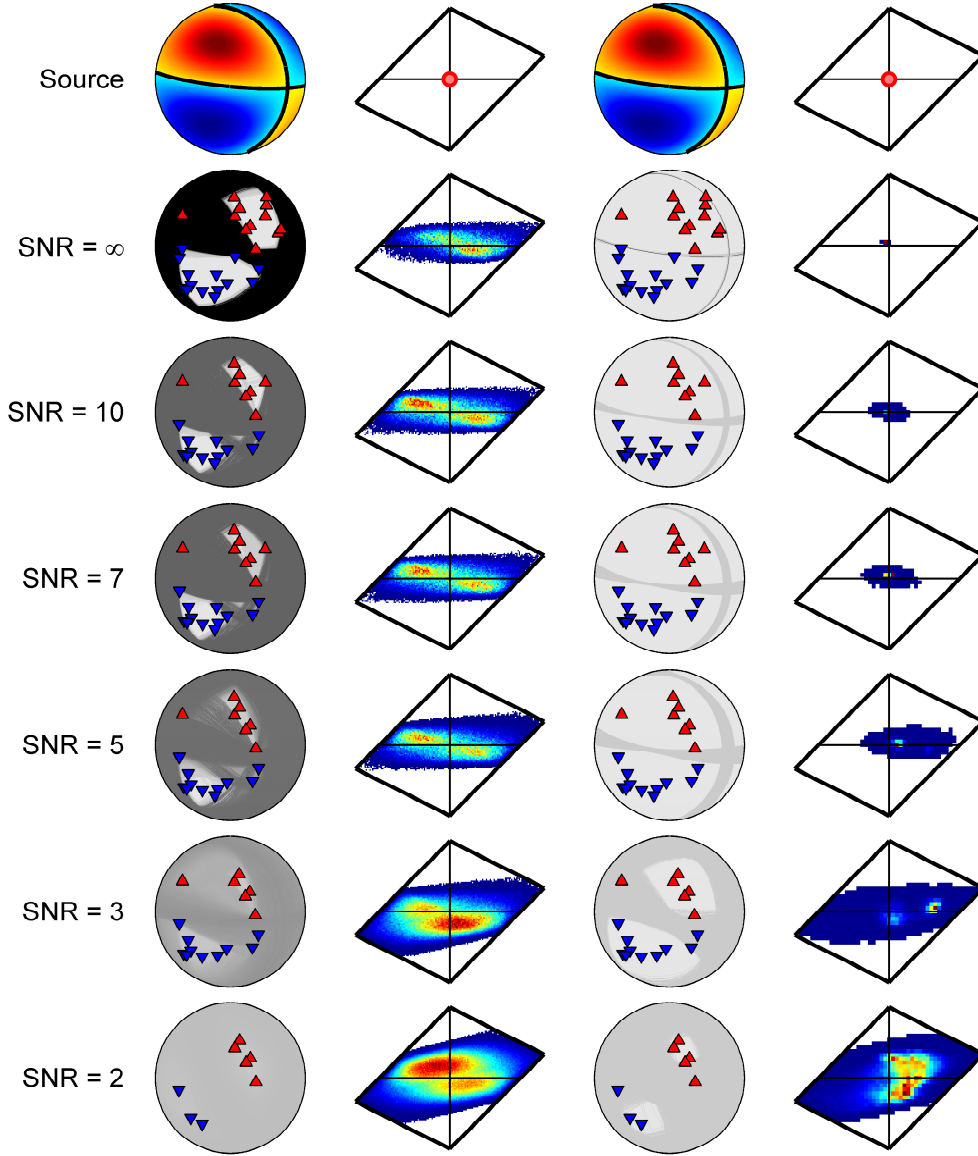


Fig. 4: Lower hemisphere equal area projections and Hudson plots of the source PDF for a synthetic double-couple source for a range of data uncertainties, corresponding to $\text{SNR} = \infty$, $\text{SNR} = 10$, $\text{SNR} = 7$, $\text{SNR} = 5$, $\text{SNR} = 3$ and $\text{SNR} = 2$. The first and third columns show the source PDF for the solution constrained to be double-couple only. The second and fourth columns show the source PDF for the full moment tensor solution. The first two columns show the solutions for inversions using only polarity data, and the second two columns show the solutions using polarity and amplitude ratio data. Manually picked station first motions are given by upward red or downward blue triangles. For the focal sphere plots, possible fault planes are given by dark lines. The most likely fault planes are given by the darkest lines. For the Hudson plots, high probability is red and low probability is in blue.

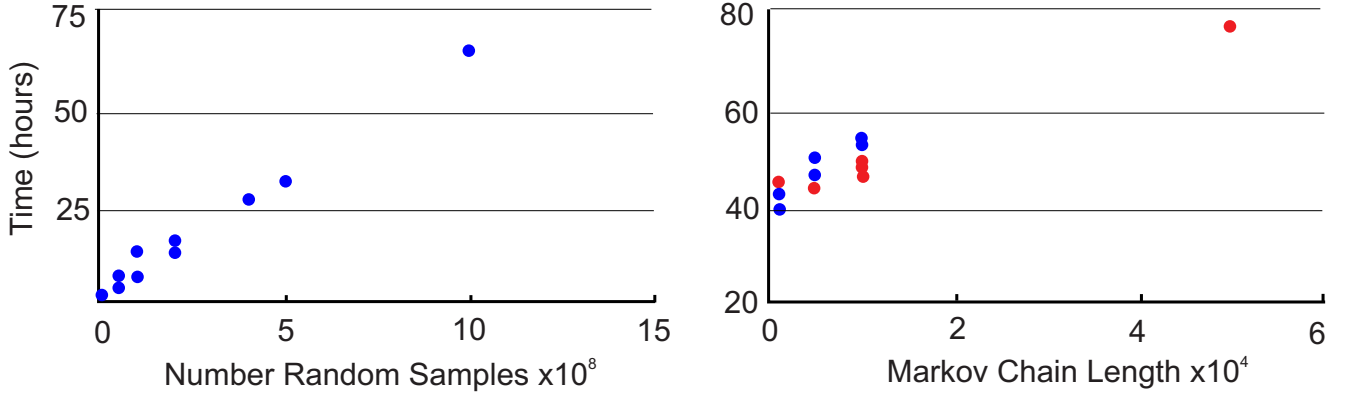


Fig. 6: Elapsed time on a single core computer for different sample sizes of the random sampling algorithm (left plot) and for the MCMC algorithms with different chain lengths (right plot) for a double couple source which includes location and velocity model uncertainties. The red dots in the MCMC case correspond to the trans-dimensional MCMC algorithm and the blue dots correspond to the standard algorithm. The velocity model and location uncertainty in the source was included with a one degree binning, reducing the number of location samples from 50,000 to 5,463.

188 Extending MTfit

189 MTfit has been written so that it is easy to extend. This is achieved using the Python setuptools
 190 module (<https://pythonhosted.org/setuptools/>), which provides entry points for a module.
 191 These entry points enable a module to check for other functions in different modules that have
 192 been advertised at this entry point, and can call them without any changes to the source code of
 193 either module. The MTfit documentation provides a more comprehensive description of the
 194 entry points, and how to call them, but a small overview is provided here.

195 Table 1 shows the list of entry points for MTfit. This section presents a step-by-step guide for
 196 installing an example data parser entry point.

197 First, the parser code must be written, which requires understanding the format of the input
 198 data, and parsing the required observations to be used in MTfit. The return data format is a
 199 Python dictionary of data per event, with the results for multiple events corresponding to a list
 200 of dictionaries.

201 https://github.com/djpugh/MTfit/tree/master/examples/SRL_examples/simple_parser.py

Entry Point	Description
<i>MTfit.cmd_opts</i>	Command line options
<i>MTfit.cmd_defaults</i>	Default parameters for the command line options
<i>MTfit.tests</i>	Test functions for the extensions
<i>MTfit.pre_inversion</i>	Function to be called with all kwargs before the inversion object is initialised
<i>MTfit.post_inversion</i>	Function to be called with all available kwargs after the inversion has occurred
<i>MTfit.extensions</i>	Functions that replace the call to the inversion using all the kwargs
<i>MTfit.parsers</i>	Functions that return the data dictionary from an input filename
<i>MTfit.location_pdf_parsers</i>	Functions that return the location PDF samples from an input filename
<i>MTfit.output_data_formats</i>	Functions that format the output data into a given type, often linked to the output format
<i>MTfit.output_formats</i>	Functions that output the results from the output_data_formats
<i>MTfit.process_data_types</i>	Functions to convert input data into correct format for new data types in forward model
<i>MTfit.data_types</i>	Functions to evaluate the forward model for new data types
<i>MTfit.parallel_algorithms</i>	Search algorithms that can be run (in parallel) like MC random sampling
<i>MTfit.directed_algorithms</i>	Search algorithms that are dependent on the previous value (e.g., McMC)
<i>MTfit.sampling</i>	Function that generates new moment tensor samples in the MC random sampling algorithm
<i>MTfit.sampling_prior</i>	Function that calculates the prior probability distribution either in the McMC algorithm or the MC Bayesian evidence estimate
<i>MTfit.sample_models</i>	Function that generates random samples according to some source model
<i>MTfit.plot</i>	Callable class for source plotting using matplotlib
<i>MTfit.plot_read</i>	Function that reads the data from a file for the MTplot class
<i>MTfit.documentation</i>	Installs the documentation for the extension
<i>MTfit.source_code</i>	Installs the source code documentation for the extension

Tab. 1: List of MTfit entry points and their short descriptions. For details see the MTfit documentation.

202 shows an example parser for a simple data format of

203 ReceiverName\tPolarity\tError\tAzimuth\tTakeOffAngle.

204 This parser can be installed using the MTfit.parsers entry point, which requires a setuptools

205 setup.py file for the parser, which should contain the entry point definition:

206 `kwargs['entry_points'] = {'MTfit.parsers': ['.sim = example:simple_parser']}`

207 With the parser installed, input files that end in .sim can be read by MTfit.

208 Similar approaches for the other entry points allow further extension of MTfit.

209 Plotting Results

210 MTfit also has a plotting submodule, MTfit.plot, which uses matplotlib

211 (<https://www.matplotlib.org>) to plot the results. It can handle several different plot types,

212 including beachball plots, fault plane plots, Riedesel-Jordan plots (Riedesel & Jordan, 1989),

213 radiation pattern plots, lune plots (Tape & Tape, 2012), and Hudson plots (Hudson *et al.*, 1989).

214 These are shown in Figure 7, which also shows several representations of the source PDF on the

215 fault plane, lune, and Hudson plots. The MTfit.plot entry point allows other plot types to be

216 added easily.

217 An example script for generating the plots in Figures 2 and 7 is shown in

218 https://github.com/djpugh/MTfit/tree/master/examples/SRL_examples/plot_krafla.py

219 There is a similar MATLAB® module, MTplot, available from

220 <https://github.com/djpugh/MTplot>, which can produce similar plot types and also several

221 additional ones.

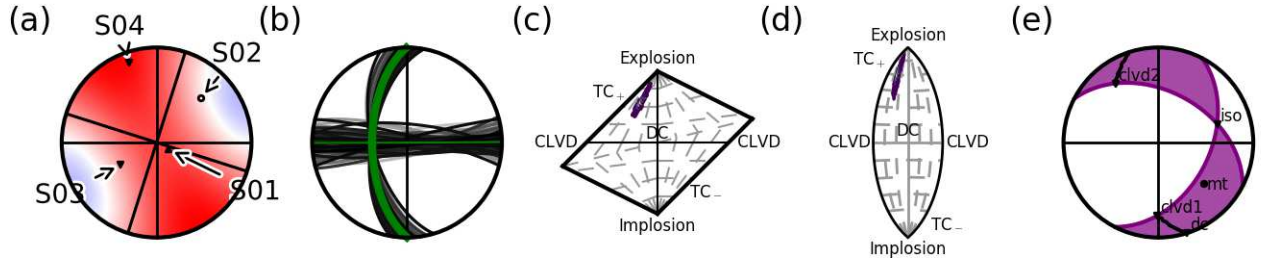


Fig. 7: MTplot examples showing (a) an equal area projection of a beachball for an example moment tensor source, (b) fault plane distribution showing the mean orientation in green, (c) Hudson and (d) lune type plots of a full moment tensor PDF, and (e) a Riedesel-Jordan type plot of an example moment tensor source.

Conclusion

MTfit is a Python module for Bayesian source inversion using different data types. It has been written to allow easy extension using Python and C modules. It has an in-built test suite, which allows changes to the code base to be tested, and it is platform independent, requiring only Python. It has been written to take advantage of parallel computation, both on a single machine and over a larger cluster, using MPI and multiprocessing.

MTfit provides an easily accessible and extendable updated approach to source inversion. The detailed documentation and package can be accessed at <https://github.com/djpugh/MTfit>.

Data and Resources

The example data used here are included in the MTfit package and have been published in Mildon *et al.* (2016). The MTfit package and detailed documentation is available from <https://github.com/djpugh/MTfit> for research and teaching i.e. for non-commercial use only.

The methods incorporated into the MTfit package are patents-pending, protected, and licensed intellectual property. Applications for commercial use of the MTfit package and/or its underlying

236 methodologies should be made to either Schlumberger or Cambridge Enterprise Limited.

237 **Acknowledgments**

238 The MTfit code was developed as part of a PhD project (Pugh 2015), funded under a Natural
239 Environment Research Council (NERC) studentship at the University of Cambridge, as a CASE
240 award with Schlumberger. Seismometers were borrowed from the NERC SEIS-UK (loan 842). We
241 thank those who have helped, advised, and tested releases of MTfit during its development, in-
242 cluding Mike Williams, Phil Christie, Matt Wilks, Tim Greenfield, Rob Green, Jenny Woods, and
243 Thorbjorg Agustsdottir, among others. Department of Earth Sciences, Cambridge contribution
244 number esc.4083.

245 References

- 246 Beyreuther, M., Barsch, R., Krischer, L., Megies, T., Behr, Y., & Wassermann, J., 2010. ObsPy:
247 a Python toolbox for seismology, *SRL: Electronic Seismologist*, **81**, 530–533.
- 248 Greenfield, T. & White, R. S., 2015. Building Icelandic igneous crust by repeated melt injections,
249 *Journal of Geophysical Research*, **120**, doi: 10.1002/2015JB012009.
- 250 Hardebeck, J. L. & Shearer, P. M., 2002. A new method for determining first-motion focal mech-
251 anisms, *Bulletin of the Seismological Society of America*, **92**(6), 2264–2276.
- 252 Hardebeck, J. L. & Shearer, P. M., 2003. Using S / P amplitude ratios to constrain the focal
253 mechanisms of small earthquakes, *Bulletin of the Seismological Society of America*, **93**(6), 2434–
254 2444.
- 255 Hudson, J. A., Pearce, R. G., & Rogers, R. M., 1989. Source type plot for inversion of the moment
256 tensor, *Journal of Geophysical Research*, **94**(B1), 765–774.
- 257 Hudson, T. S., White, R. S., Greenfield, T., Agustsdottir, T., Brisbourne, A., & Green, R. G., 2017.
258 Deep crustal melt plumbing of Bardarbunga volcano, Iceland, *Geophysical Research Letters*, **44**,
259 3710–3718, doi:10.1002/2017GL074749.
- 260 Lomax, A., Virieux, J., Volant, P., & Berge, C., 2000. *Probabilistic earthquake location in 3D and*
261 *layered models: Introduction of a Metropolis-Gibbs method and comparison with linear locations*,
262 pp. 101–134, *Advances in Seismic Location*, Kluwer.
- 263 Lomax, A., Michelini, A., & Curtis, A., 2009. *Earthquake Location, Direct, Global-Search Methods*,
264 pp. 2449–2473, *Encyclopedia of Complexity and System Science*, Part 5, Springer.
- 265 Mildon, Z. K., Pugh, D. J., Tarasewicz, J., White, R. S., & Brandsdóttir, B., 2016. Closing crack

266 earthquakes within the Krafla caldera, North Iceland, *Geophysical Journal International*, **207**,
267 1137–1141.

268 Pugh, D. J., 2015. *Bayesian Source Inversion of Microseismic Events*, Ph.D. thesis, University of
269 Cambridge.

270 Pugh, D. J., White, R. S., & Christie, P. A. F., 2016a. Automatic Bayesian polarity determination,
271 *Geophysical Journal International*, **206**(1), 275–291.

272 Pugh, D. J., White, R. S., & Christie, P. A. F., 2016b. A Bayesian method for microseismic source
273 inversion, *Geophysical Journal International*, **206**(2), 1009–1038.

274 Reasenber, P. A. & Oppenheimer, D., 1985. FPFIT, FPLOT and FPPAGE: Fortran computer
275 programs for calculating and displaying earthquake fault-plane solutions - OFR 85-739, Tech.
276 rep., USGS.

277 Riedesel, M. A. & Jordan, T. H., 1989. Display and assessment of seismic moment tensors, *Bulletin*
278 *of the Seismological Society of America*, **79**(1), 85–100.

279 Schuler, J., Pugh, D. J., Hauksson, E., White, R. S., Stock, J. M., & Brandsdóttir, B., 2016.
280 Focal mechanisms and size distribution of earthquakes beneath the Krafla central volcano, NE
281 Iceland, *Journal of Geophysical Research: Solid Earth*, **121**(7), 5152–5168.

282 Smith, E. C., Baird, A., Kendall, J. M., Martin, C., White, R. S., Bourne, A. M., & Smith,
283 A. M., 2017. Ice Fabric in an Antarctic ice stream interpreted from seismic anisotropy, *Geo-*
284 *physical Research Letters*, **44**, 3710–3718, doi:10.1002/2016GL072093.

285 Snoke, J. A., 2003. FOCMEC: FOCal MEChanism determinations, Tech. rep., Virginia Tech.

286 Tape, W. & Tape, C., 2012. A geometric setting for moment tensors, *Geophysical Journal Inter-*
287 *national*, **190**(1), 476–498.

288 Wilks, M., Bradford, I., Williams, M., Rodriguez, I. V., & Pugh, D., 2015. Combined use of a
289 deep monitoring array and shallow borehole arrays for moment tensor inversion: Th N108 09,
290 in *77th EAGE Conference and Exhibition*.

291 Figure Captions

292 Figure 1. Flow diagram outlining the main steps in the moment tensor inversion package.

293

294 Figure 2. Krafla example results from the script at

295 https://github.com/djpugh/MTfit/tree/master/examples/SRL_examples/krafla.py (run with

296 1,000,000 samples). The first plot shows the station distribution of observed receivers on the

297 focal sphere, all with negative polarity, determined from the NonLinLoc estimate of the location

298 PDF. The lighter points correspond to more likely receiver locations, and the maximum

299 likelihood station locations with observed polarities are shown as triangles. The second plot

300 shows the fault plane distribution for the double-couple constrained solution, with darker fault

301 planes more likely. The last plot shows the Hudson type plot of the marginalized source-type

302 PDF from the full moment tensor solution, with dark regions corresponding to low-probability

303 source-types and lighter areas to higher probability types.

304

305 Figure 3. Marginalised posterior parameter distribution histogram for the five parameters

306 described in Tape & Tape (2012) for the event shown in Fig. 2. γ and δ describe the moment

307 tensor pattern, while κ (strike angle), h (cosine of dip) and σ (rake) describe the orientation. All

308 parameters are dimensionless except κ and σ , which are in radians. This shows that the

309 distributions are well constrained for the δ component, but are less well constrained for the fault

310 plane orientation and γ components.

311

312 Figure 4. Lower hemisphere equal area projections and Hudson plots of the source PDF for a

313 synthetic double-couple source for a range of data uncertainties, corresponding to SNR =

314 infinity, SNR = 10, SNR = 7, SNR = 5, SNR = 3 and SNR = 2. The first and third columns

show the source PDF for the solution constrained to be double-couple only. The second and fourth columns show the source PDF for the full moment tensor solution. The first two columns show the solutions for inversions using only polarity data, and the second two columns show the solutions using polarity and amplitude ratio data. Manually picked station first motions are given by upward red or downward blue triangles. For the focal sphere plots, possible fault planes are given by dark lines. The most likely fault planes are given by the darkest lines. For the Hudson plots, high probability is red and low probability is in blue.

322

Figure 5. Elapsed time on a single core computer for different sample sizes of the random sampling (left plot) and for the McMC algorithms with different chain lengths (right plot) for a double couple source with no uncertainties in the input data. The red dots in the McMC case correspond to the trans-dimensional McMC algorithm and the blue dots correspond to the standard algorithm.

328

Figure 6. Elapsed time on a single core computer for different sample sizes of the random sampling algorithm (left plot) and for the McMC algorithms with different chain lengths (right plot) for a double couple source which includes location and velocity model uncertainties. The red dots in the McMC case correspond to the trans-dimensional McMC algorithm and the blue dots correspond to the standard algorithm. The velocity model and location uncertainty in the source was included with a one degree binning, reducing the number of location samples from 50,000 to 5,463.

336

Figure 7. MTplot examples showing (a) an equal area projection of a beachball for an example moment tensor source, (b) fault plane distribution showing the mean orientation in green, (c)

338

339 Hudson and (d) lune type plots of a full moment tensor PDF, and (e) a Riedesel-Jordan type
340 plot of an example moment tensor source.